UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/849,736	05/04/2001	G. Glenn Henry	CNTR:2021	8084

23669 7590 08/31/2004

HUFFMAN LAW GROUP, P.C.
1832 N. CASCADE AVE.
COLORADO SPRINGS, CO 80907-7449

EXAMINER	
HUISMAN, DAVID J	

ART UNIT	PAPER NUMBER
2183	

DATE MAILED: 08/31/2004

Please find below and/or attached an Office communication concerning this application or proceeding.

Office Action Summary	Application No.	Applicant(s)	
	09/849,736	HENRY ET AL.	
	Examiner	Art Unit	
	David J. Huisman	2183	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If the period for reply specified above is less than thirty (30) days, a reply within the statutory minimum of thirty (30) days will be considered timely.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *08 June 2004*.

2a)☐ This action is **FINAL**. 2b)☒ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1,3-11,13-18,31-38 and 43-70* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1,3-11,13-18,31-38 and 43-70* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☒ The specification is objected to by the Examiner.

10)☒ The drawing(s) filed on *04 May 2001* is/are: a)☒ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☐ Information Disclosure Statement(s) (PTO-1449 or PTO/SB/08) Paper No(s)/Mail Date _____.
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____.
5) ☐ Notice of Informal Patent Application (PTO-152)
6) ☐ Other: _____.

DETAILED ACTION

1. Claims 1, 3-11, 13-18, 31-38, and 43-70 have been examined.

Papers Submitted

2. It is hereby acknowledged that the following papers have been received and placed of record in the file: Amendment as received on 6/8/2004 and Oath/Declaration as received on 6/15/2004.

Specification

3. The title of the invention is not descriptive. A new title is required that is clearly indicative of the invention to which the claims are directed.

Claim Objections

4. Claim 48 is objected to because of the following informalities: In line 2, insert either --the-- or --said-- before "instruction buffer". Appropriate correction is required.

Withdrawn Rejections

5. Applicant's arguments on pages 22-23 of the remarks field on June 8, 2004, regarding claims 12 and 16 are deemed persuasive by the examiner. Consequently, the rejections have been withdrawn and a new ground(s) of rejection is made below.

Claim Rejections - 35 USC § 112

6. The following is a quotation of the second paragraph of 35 U.S.C. 112:

> The specification shall conclude with one or more claims particularly pointing out and distinctly claiming the subject matter which the applicant regards as his invention.

7. Claims 48-49 are rejected under 35 U.S.C. 112, second paragraph, as being indefinite for failing to particularly point out and distinctly claim the subject matter which applicant regards as the invention. More specifically, these claims are associated with non-speculative predictions. The examiner asserts that this is unclear because it is well known that when one predicts something, one is also speculating that something will occur. It is not understood how a prediction may be non-speculative.

Claim Rejections - 35 USC § 103

8. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

9. Claims 1, 3-7, 11, 13-18, 34-38, 50-55, and 59-70 are rejected under 35 U.S.C. 103(a) as being unpatentable over Shiell et al., U.S. Patent No. 5,850,543 (as applied in the previous Office Action and herein referred to as Shiell) in view of Hsu et al., U.S. Patent No. 5,948,100 (herein referred to as Hsu).

10. Referring to claim 1, Shiell has taught a branch target address cache (BTAC) (fig. 2, Branch Target Buffer, BTB 56) for providing a speculative target address (col. 7, lines 40-42; the target address is speculative because it is provided during the instruction fetch stage before it is known whether the instruction is a branch or not) to address selection logic (fig. 2, multiplexers 57, 58 and 52), the address selection logic selecting a fetch

address for addressing a line in an instruction cache (fig. 2 shows that the address selection logic addresses the instruction cache 16_i and col. 6, lines 16-23 indicate that the fetch address addresses a stream of instruction data i.e. a line in the instruction cache), the BTAC providing the speculative target address based on a presumption that a branch instruction is present in the cache (fig. 2 shows that the BTB outputs a target address D0-D127 indexed by the fetch address without knowing whether or not a branch is present in the line of the instruction cache addressed by the fetch address because the BTB and the instruction cache are accessed in parallel using the fetch address FA), the BTAC comprising:

a) an array of storage elements, configured to cache target addresses of previously executed branch instructions (fig. 2, BTB 56; col. 7, lines 40-42).

b) Shiell has not taught storing speculative branch information associated with said previously executed branch instructions, wherein said speculative branch information comprises a length of the branch instruction presumed present in the cache line. However, Hsu has taught a BTB which caches the length of each branch instruction. See column 13, lines 18-30. This allows the system to accommodate branch instructions that cross multiple cache blocks, which is useful in variable length instruction systems, as both Hsu and Shiell are (see the title of Hsu and column 6, lines 32-35 of Shiell). Consequently, it would have been obvious to one of ordinary skill in the art at the time of the invention to modify Shiell to cache the length of a branch instruction so that the system may accommodate branches that span multiple cache lines.

c) an input, coupled to said array, for receiving the fetch address, to index into said array of storage elements to select one of said target addresses (fig. 2; col. 8, lines 12-16).

d) an output, coupled to said array, for providing said one of said target addresses indexed by the fetch address to the address selection logic (fig. 2 shows the target address D0-D127 are connected to an output to the address selection logic [multiplexers 57, 58 and 52]);

e) wherein said output provides said one of said target addresses to the address selection logic for selection as a subsequent fetch address whether or not a branch instruction is present in the line of the instruction cache addressed by the fetch address (fig. 2 shows that the BTB outputs a target address D0-D127 indexed by the fetch address to the address selection logic (multiplexers 57, 58 and 52) without knowing whether or not a branch is present in the line of the instruction cache addressed by the fetch address because the BTB and the instruction cache are accessed in parallel using the fetch address FA).

11. Referring to claim 3, Shiell in view of Hsu has taught a BTAC as described in claim 1. Shiell has further taught a second output, coupled to said array, for providing a portion of said speculative branch information to control logic for controlling the address selection logic in response to said portion of said speculative branch information (although not shown, it is deemed inherent to the BTB to have a second output which outputs a portion of the history information indicating the outcome of the branch to control the address selection logic multiplexers. This is because when the branch is indicated in the BTB as not taken, the BTB target address should not be selected and the next sequential address should be selected).

12. Referring to claim 4, Shiell in view of Hsu has taught a BTAC as described in claim 1. Shiell has further taught that said speculative branch information comprises

information predicting whether the branch instruction presumed present in the cache line will be taken (fig. 3, "HIS" field indicates that the branch will be taken if set to '111' or '110' [col. 8, lines 57-67]).

13. Referring to claim 5, Shiell in view of Hsu has taught a BTAC as described in claim 4. Shiell has further taught that said information predicting whether the presumed branch instruction will be taken comprises a taken/not taken bit (fig. 9, when a conditional branch, the 2nd bit in the "HIS" field is a taken/not taken bit because when set to "1" it indicates taken, when set to "0" it indicates not taken).

14. Referring to claim 6, Shiell in view of Hsu has taught a BTAC as described in claim 4. Shiell has further taught that said information predicting whether the presumed branch instruction will be taken comprises a plurality of bits (fig. 3, "HIS" field).

15. Referring to claim 7, Shiell in view of Hsu has taught a BTAC as described in claim 6. Shiell has further taught that said plurality of bits is stored in a saturating up/down counter (col. 9, lines 5-18).

16. Referring to claim 11, Shiell in view of Hsu has taught a BTAC as described in claim 1. Shiell has further taught that said speculative branch information comprises information specifying a location within the cache line of the branch instruction presumed present in the cache line (fig. 3, shows the "T_n" field as an entry in the BTB. col. 8, lines 40-44 indicate that the T_n field holds information specifying the location of a specific instruction within the cache line associated with the logical address LA used to index into the BTB).

17. Referring to claim 13, Shiell in view of Hsu has taught a BTAC as described in claim 1. Shiell has further taught that said speculative branch information comprises an

indication of a type of the branch instruction presumed present in the cache line (fig. 2, "HIS" field, col. 8, lines 57-67).

18. Referring to claim 14, Shiell in view of Hsu has taught a BTAC as described in claim 13. Shiell has further taught that said indication of said type of the branch instruction indicates whether the branch instruction is a call instruction ("HIS" = 011; col. 8, lines 57-67).

19. Referring to claim 15, Shiell in view of Hsu has taught a BTAC as described in claim 13. Shiell has further taught that said indication of said type of the branch instruction indicates whether the branch instruction is a return instruction ("HIS" = 010; col. 8, lines 57-67).

20. Referring to claim 16, Shiell in view of Hsu has taught a BTAC as described in claim 1. Hsu has further taught that said speculative branch information comprises an indication of the branch instruction presumed present in the cache line spans more than one line in the instruction cache. See column 13, lines 18-30, and recall that Hsu has taught a BTB which caches the length of each branch instruction. The length is used to determine whether or not a branch spans a cache line by adding it to an address. Therefore, the length is at the very least an indirect or partial indication.

21. Referring to claim 17, Shiell in view of Hsu has taught a BTAC as described in claim 1. Shiell has further taught that each of said storage elements is configured to cache a plurality of target addresses (col. 8, lines 22-27 indicates that there are 4 target addresses stored per storage element).

22. Referring to claim 18, Shiell in view of Hsu has taught a BTAC as described in claim 1. Shiell has further taught that the branch target address cache is external to the instruction cache (fig. 2).

23. Referring to claim 34, Shiell has taught a method of speculatively branching in a pipelined microprocessor (fig. 1), comprising:

a) caching a plurality of branch target addresses of previously executed branch instructions in a branch target address cache (BTAC) (fig. 2, BTB 56; col. 7, lines 40-42).

b) Shiell has not taught caching an indication of whether each of the branch instructions spans an instruction cache line in the BTAC. However, Hsu has taught a BTB which caches the length of each branch instruction. The length is used to determine whether or not a branch spans a cache line by adding it to an address. See column 13, lines 18-30. Therefore, the length is at the very least an indirect or partial indication. By tracking the length, the system is able to accommodate branch instructions that cross multiple cache blocks, which is useful in variable length instruction systems, as both Hsu and Shiell are (see the title of Hsu and column 6, lines 32-35 of Shiell). Consequently, it would have been obvious to one of ordinary skill in the art at the time of the invention to modify Shiell to cache an indication of whether a branch instruction spans an instruction cache line.

c) accessing said BTAC with a fetch address of an instruction cache after said caching (fig. 2 indicates that the fetch address FA is used to access the BTB 56; col. 8, lines 13-16 indicate that the accessing is done when branch history is stored in the BTB).

d) determining whether said fetch address hits in said BTAC in response to said accessing (col. 8, lines 13-16).

e) branching the microprocessor to one of said plurality of branch target addresses selected by said fetch address if said fetch address hits in said BTAC whether or not a branch instruction is cached in a line of said instruction cache indexed by said fetch address (col. 7, lines 40-45 indicate that the target address is used to generate the fetch address during the fetch stage before the decoding the instruction without knowledge of whether the instruction at that fetch address in the instruction cache is still a branch or not i.e. speculative execution [col. 8, lines 13-16] because the BTB and the instruction cache are accessed in parallel using the fetch address FA).

24. Referring to claim 35, Shiell in view of Hsu has taught a method as described in claim 34. Shiell has further taught storing a branch direction prediction associated with each of said plurality of branch target addresses prior to said accessing said BTAC (fig. 3, "HIS_n" field; col. 8, lines 57-67).

25. Referring to claim 36, Shiell in view of Hsu has taught a method as described in claim 35. Shiell has further taught that said branching the microprocessor to said one of said plurality of branch target addresses selected by said fetch address is performed only if said associated branch direction prediction indicates said branch instruction will be taken (Although not explicitly mentioned, the limitation is deemed inherent to the correct functioning of the method because the purpose of the prediction information when indicating that the speculative branch is taken is for instructing the processor to branch to the target address of the branch and not the next sequential address).

26. Referring to claim 37, Shiell in view of Hsu has taught a method as described in claim 34. Shiell has further taught storing an indication said branching was performed if said branching is performed (col. 7, lines 40-45, 65-67; col. 8, lines 1-4 indicate that if

branching is performed by using the target address as the fetch address, the instruction from the branch target path is fetched and stored in the instruction buffer hence indicating that branching is performed).

27. Referring to claim 38, Shiell in view of Hsu has taught a method as described in claim 37. Shiell has further taught that said storing said indication said branching was performed comprises storing said indication in an instruction buffer (col. 7, lines 40-45, 65-67; col. 8, lines 1-4 indicate that if branching is performed by using the target address as the fetch address, the instruction from the branch target path is fetched and stored in the instruction buffer hence indicating that branching is performed).

28. Referring to claim 50, applicant states on page 26 (in the remarks of the amendment), that claim 50 includes the combined limitations of original claims 1, 2, and 16. Consequently, claim 50 is rejected for the same reasons set forth in the rejections of claims 1 and 16 above.

29. Referring to claim 51, Shiell in view of Hsu has taught a BTAC as described in claim 50. Furthermore, claim 51 is rejected for the same reasons set forth in the rejection of claim 3.

30. Referring to claim 52, Shiell in view of Hsu has taught a BTAC as described in claim 50. Furthermore, claim 52 is rejected for the same reasons set forth in the rejection of claim 4.

31. Referring to claim 53, Shiell in view of Hsu has taught a BTAC as described in claim 52. Furthermore, claim 53 is rejected for the same reasons set forth in the rejection of claim 5.

32. Referring to claim 54, Shiell in view of Hsu has taught a BTAC as described in claim 52. Furthermore, claim 54 is rejected for the same reasons set forth in the rejection of claim 6.

33. Referring to claim 55, Shiell in view of Hsu has taught a BTAC as described in claim 54. Furthermore, claim 55 is rejected for the same reasons set forth in the rejection of claim 7.

34. Referring to claim 59, Shiell in view of Hsu has taught a BTAC as described in claim 50. Furthermore, claim 59 is rejected for the same reasons set forth in the rejection of claim 11.

35. Referring to claim 60, Shiell in view of Hsu has taught a BTAC as described in claim 50. Shiell has not taught that said speculative branch information comprises a length of the branch instruction presumed present in the cache line. However, Hsu has taught a BTB which caches the length of each branch instruction. See column 13, lines 18-30. This allows the system to accommodate branch instructions that cross multiple cache blocks, which is useful in variable length instruction systems, as both Hsu and Shiell are (see the title of Hsu and column 6, lines 32-35 of Shiell). Consequently, it would have been obvious to one of ordinary skill in the art at the time of the invention to modify Shiell to cache the length of a branch instruction so that the system may accommodate branches that span multiple cache lines.

36. Referring to claim 61, Shiell in view of Hsu has taught a BTAC as described in claim 50. Furthermore, claim 61 is rejected for the same reasons set forth in the rejection of claim 13.

37. Referring to claim 62, Shiell in view of Hsu has taught a BTAC as described in claim 61. Furthermore, claim 62 is rejected for the same reasons set forth in the rejection of claim 14.

38. Referring to claim 63, Shiell in view of Hsu has taught a BTAC as described in claim 61. Furthermore, claim 63 is rejected for the same reasons set forth in the rejection of claim 15.

39. Referring to claim 64, Shiell in view of Hsu has taught a BTAC as described in claim 50. Furthermore, claim 64 is rejected for the same reasons set forth in the rejection of claim 16.

40. Referring to claim 65, Shiell in view of Hsu has taught a BTAC as described in claim 50. Furthermore, claim 65 is rejected for the same reasons set forth in the rejection of claim 17.

41. Referring to claim 66, Shiell has taught a method of speculatively branching in a pipelined microprocessor (fig. 1), comprising:

a) caching a plurality of branch target addresses of previously executed branch instructions in a branch target address cache (BTAC) (fig. 2, BTB 56; col. 7, lines 40-42).

b) Shiell has not taught caching a length of each of the branch instructions in the BTAC. However, Hsu has taught a BTB which caches the length of each branch instruction. See column 13, lines 18-30. This allows the system to accommodate branch instructions that cross multiple cache blocks, which is useful in variable length instruction systems, as both Hsu and Shiell are (see the title of Hsu and column 6, lines 32-35 of Shiell). Consequently, it would have been obvious to one of ordinary skill in the art at the time of

the invention to modify Shiell to cache the length of a branch instruction so that the system may accommodate branches that span multiple cache lines.

c) accessing said BTAC with a fetch address of an instruction cache after said caching (fig. 2 indicates that the fetch address FA is used to access the BTB 56; col. 8, lines 13-16 indicate that the accessing is done when branch history is stored in the BTB).

d) determining whether said fetch address hits in said BTAC in response to said accessing (col. 8, lines 13-16).

e) branching the microprocessor to one of said plurality of branch target addresses selected by said fetch address if said fetch address hits in said BTAC whether or not a branch instruction is cached in a line of said instruction cache indexed by said fetch address (col. 7, lines 40-45 indicate that the target address is used to generate the fetch address during the fetch stage before the decoding the instruction without knowledge of whether the instruction at that fetch address in the instruction cache is still a branch or not i.e. speculative execution [col. 8, lines 13-16] because the BTB and the instruction cache are accessed in parallel using the fetch address FA).

42. Referring to claim 67, Shiell in view of Hsu has taught a method as described in claim 66. Furthermore, claim 67 is rejected for the same reasons set forth in the rejection of claim 35.

43. Referring to claim 68, Shiell in view of Hsu has taught a method as described in claim 67. Furthermore, claim 68 is rejected for the same reasons set forth in the rejection of claim 36.

44. Referring to claim 69, Shiell in view of Hsu has taught a method as described in claim 66. Furthermore, claim 69 is rejected for the same reasons set forth in the rejection of claim 37.

45. Referring to claim 70, Shiell in view of Hsu has taught a method as described in claim 69. Furthermore, claim 70 is rejected for the same reasons set forth in the rejection of claim 38.

46. Claims 8-10 and 56-58 are rejected under 35 U.S.C. 103(a) as being unpatentable over Shiell in view of Hsu, as applied above, and further in view of Bae et al., U.S. Patent No. 6,044,459 (as applied in the previous Office Action and herein referred to as Bae).

47. Referring to claim 8, Shiell in view of Hsu has taught a BTAC as described in claim 3. Furthermore, although Shiell verifies whether the target address was correct by comparing it with the actual next instruction address determined by the execution unit (col. 2, lines 48-51), Shiell has not taught that said portion of said speculative branch information comprises an indication of whether said one of said target addresses is a valid target address. However, Bae has taught a BTB entry format with a valid bit for indicating whether the target address is a valid target address (col. 5, lines 27-34). The valid bit is checked before providing the target address as the fetch address so that an invalid target is not incorrectly provided. It would have been obvious to one of ordinary skill in the art at the time of the invention to recognize to add a valid bit in the BTB of Shiell indicating the validity of the target address so that in invalid target is not provided. One would have been motivated to do so because by including the valid bit, incorrect

targets are not fetched and subsequent flushing of the pipeline is not required, thereby leading to improved performance.

48. Referring to claim 9, Shiell in view of Hsu and further in view of Bae has taught a BTAC as described in claim 8. Shiell in view of Hsu and further in view of Bae has further taught that said indication is populated to indicate said one of said target addresses is a valid target address in response to execution of the presumed branch instruction, wherein said one of said target addresses is resolved. This limitation is deemed inherent because the valid bit can be set to indicate the validity of the predicted target address only after the branch instruction is executed and the target is calculated.

49. Referring to claim 10, Shiell in view of Hsu and further in view of Bae has taught a BTAC as described in claim 8. Shiell in view of Hsu and further in view of Bae has further taught that said indication is populated to indicate said one of said target addresses is not a valid target address in response to detecting said one of said target addresses is erroneous subsequent to said providing said one of said target addresses on said output (This limitation is deemed inherent because the valid bit can be set to indicate that the predicted target address is invalid only after comparing the predicted target address in the BTB with the target address calculated on execution i.e. detecting that the target address is erroneous).

50. Referring to claim 56, Shiell in view of Hsu has taught a BTAC as described in claim 51. Furthermore, claim 56 is rejected for the same reasons set forth in the rejection of claim 8.

51. Referring to claim 57, Shiell in view of Hsu has taught a BTAC as described in claim 56. Furthermore, claim 57 is rejected for the same reasons set forth in the rejection of claim 9.

52. Referring to claim 58, Shiell in view of Hsu has taught a BTAC as described in claim 56. Furthermore, claim 58 is rejected for the same reasons set forth in the rejection of claim 10.

53. Claims 31-33, 43-44, and 46-49 are rejected under 35 U.S.C. 103(a) as being unpatentable over Shiell, as applied above, in view of Dietz et al., U.S. Patent No. 5,634,103 (as applied in the previous Office Action and herein referred to as Dietz).

54. Referring to claim 31, Shiell has taught a pipelined microprocessor comprising:

a) an instruction cache (fig. 2, 16_i) that is indexed by a fetch address, said instruction cache for caching instructions (col. 5, lines 37-38), and for providing said instructions to an instruction buffer (fig 2, 60).

b) a branch target address cache (fig. 2, BTB 56), coupled to said instruction buffer and indexed by said fetch address (fig. 2), for caching branch target addresses of previously executed branch instructions (col. 7, lines 40-42).

c) Shiell has not taught that the instruction buffer comprises a plurality of indicators that are associated with said instructions, said indicators specifying whether the microprocessor has speculatively branched to one of said branch target addresses. However, Dietz has taught a processor in which each instruction has a corresponding speculative bit indicating whether the instruction is within the speculative execution path. If the speculative path is resolved as incorrect, the instruction tagged as speculative are

flushed (col. 4, lines 60-67). One of ordinary skill in the art would have recognized that by adding these indicators to the instructions in the instruction buffer indicating whether they are on the speculatively branched path or not, the process of flushing the speculative instructions on a misprediction simply involves flushing the instructions tagged as speculative. Moreover, Shiell mentions that instructions along the speculative path must be flushed on determining that the prediction was not correct. Hence it would have been obvious to one of ordinary skill in the art at the time of the invention to modify the processor of Shiell by adding a plurality of indicators to each instruction of the instruction buffer indicating whether the processor speculatively branched. One would have been motivated to do so because it would simplify identifying the instructions in the pipeline that are speculative and those that are not speculative while flushing the pipeline in the case of a misprediction leading to simpler logic design.

55. Referring to claim 32, Shiell in view of Dietz has taught a microprocessor as described in claim 31. Dietz has further taught that said instruction buffer includes one of said plurality of indicators associated with each byte of said each of said instructions stored in said instruction buffer (Dietz et al. teaches the use of the hit indicator for each instruction [col. 4, lines 60-65]).

56. Referring to claim 33, Shiell in view of Dietz has taught a microprocessor as described in claim 31. Shiell has further taught that said instruction cache and said branch target address cache are accessed substantially in parallel (Shiell et al. teaches that the fetch address is presented to various functions to control the fetching of the next instruction to be decoded [col. 7, lines 53-55]. fig. 2, col. 7, lines 59-62, and col. 8, lines

13-16 indicate that both the instruction cache and the BTB are accessed substantially in parallel).

57. Referring to claim 43, Shiell in view of Dietz has taught a microprocessor as described in claim 32. Furthermore, Shiell in view of Dietz has not explicitly taught instruction decode logic (see Fig.1 of Shiell), coupled to said instruction buffer, for decoding said instructions to indicate which byte of each of said instructions is an opcode byte, and prediction check logic, coupled to receive said plurality of indicators from said instruction buffer, wherein if one of said plurality of indicators associated with one of said instructions specifies the microprocessor has speculatively branched to said one of said branch target addresses of said one of said instructions, said prediction check logic indicates that the microprocessor erroneously branched to said one of said branch target addresses if a same byte of said one of said instructions is not indicated by said instruction decode logic and by said plurality of indicators stored in said instruction buffer to be said opcode byte. However, Official Notice is taken that it is common for caches (BTAC/BTB included) to be constructed such that multiple fetch addresses map to the same cache entry. If in the situation where a non-opcode byte is encountered and it maps to a branch entry that is allocated to a branch opcode, then a prediction should not be made because it is unknown whether this non-opcode byte is actually associated with a branch instruction. But, since it maps to the same location as a branch, the system will assume a branch has been encountered and an erroneous prediction would have been made. Such a cache construction is desired because it is not often that instructions map to the same location and it also keeps the cache smaller. Consequently, it would have been obvious to one of ordinary skill in the art at the time of the invention to reduce the

size of the BTAC and determine erroneous predictions if a non-opcode byte results in a prediction being made.

58. Referring to claim 44, Shiell in view of Dietz has taught a microprocessor as described in claim 31. Furthermore, Shiell in view of Dietz has not explicitly taught instruction decode logic (Fig.1 of Shiell), coupled to said instruction buffer, for decoding said instructions to indicate whether each of said instructions is a non-branch instruction, and prediction check logic, coupled to receive said plurality of indicators from said instruction buffer, wherein if one of said plurality of indicators associated with one of said instructions specifies the microprocessor has speculatively branched to said one of said branch target addresses of said one of said instructions, said prediction check logic indicates that the microprocessor erroneously branched to said one of said branch target addresses if said instruction decode logic indicates said one of said instructions is a non-branch instruction. However, Official Notice is taken that it is common for caches (BTAC/BTB included) to be constructed such that multiple fetch addresses map to the same cache entry. If in the situation where a non-branch instruction is encountered and it maps to a branch entry that is allocated to a branch instruction, then a prediction should not be made because non-branch instructions do not require predictions. But, since it maps to the same location as a branch, the system will assume a branch has been encountered and an erroneous prediction would have been made. Such a cache construction is desired because it is not often that instructions map to the same location and it also keeps the cache smaller. Consequently, it would have been obvious to one of ordinary skill in the art at the time of the invention to reduce the size of the BTAC and

determine erroneous predictions if a non-branch instruction results in a prediction being made.

59. Referring to claim 46, Shiell in view of Dietz has taught a microprocessor as described in claim 31. Furthermore, Shiell in view of Dietz has taught prediction check logic, coupled to receive said plurality of indicators from said instruction buffer, wherein said one of said instructions is a branch instruction, wherein if one of said plurality of indicators associated with said branch instruction specifies the microprocessor has speculatively branched to said one of said branch target addresses of said branch instruction, said prediction check logic indicates that the microprocessor erroneously branched to said one of said branch target addresses if a resolved direction of said branch instruction does not match a direction of said branch instruction predicted by said branch target address cache. This is deemed to be inherent because if the resolved direction of a branch does not match the predicted direction, then a misprediction has occurred (erroneously branched).

60. Referring to claim 47, Shiell in view of Dietz has taught a microprocessor as described in claim 31. Shiell in view of Dietz has further taught prediction check logic, coupled to receive said plurality of indicators from said instruction buffer, wherein said one of said instructions is a branch instruction, wherein if one of said plurality of indicators associated with said branch instruction specifies the microprocessor has speculatively branched to said one of said branch target addresses of said branch instruction, said prediction check logic indicates that the microprocessor erroneously branched to said one of said branch target addresses if a resolved target address of said branch instruction does not match said one of said branch target addresses to which the

microprocessor speculatively branched. This is deemed to be inherent because if the resolved target address of a branch does not match the speculative target address, then a misprediction has occurred (erroneously branched).

61. Referring to claim 48, Shiell in view of Dietz has taught a microprocessor as described in claim 31. Shiell in view of Dietz has further taught a non-speculative branch predictor, coupled to instruction buffer, for generating a non-speculative predicted target address of a branch instruction for which said branch target address cache provided said one of said branch target addresses to which the microprocessor speculatively branched and branch control logic, coupled to receive said plurality of indicators from said instruction buffer, wherein if one of said plurality of indicators associated with said branch instruction specifies the microprocessor has speculatively branched to said one of said branch target addresses of said branch instruction, said branch control logic causes the microprocessor to branch to said non-speculative predicted target address if said non-speculative predicted target address generated by said non-speculative branch predictor does not match said one of said branch target addresses of said branch instruction provided by said branch target address cache. This is deemed to be inherent because if the resolved target address (non-speculative prediction) of a branch does not match the actual prediction from the BTAC, then a misprediction has occurred (erroneously branched), and the system will be corrected by branching to the non-speculative address.

62. Referring to claim 49, Shiell in view of Dietz has taught a microprocessor as described in claim 31. Shiell in view of Dietz has further taught a non-speculative branch predictor, coupled to instruction buffer, for generating a non-speculative predicted direction of a branch instruction for which said branch target address cache provided said

one of said branch target addresses to which the microprocessor speculatively branched, and branch control logic, coupled to receive said plurality of indicators from said instruction buffer, wherein if one of said plurality of indicators associated with said branch instruction specifies the microprocessor has speculatively branched to said one of said branch target addresses of said branch instruction, said branch control logic causes the microprocessor to branch to a next instruction sequential to said branch instruction if said non-speculative predicted direction generated by said non-speculative branch predictor is a not taken prediction. This is deemed to be inherent because if the resolved target address (non-speculative prediction or actual target address) of a branch does not match the actual prediction from the BTAC, then a misprediction has occurred (erroneously branched), and the system will be corrected by branching to the non-speculative address. When the actual target address is the next sequential instruction, then the microprocessor will begin fetching from the next sequential address.

63. Claim 45 is rejected under 35 U.S.C. 103(a) as being unpatentable over Shiell in view of Dietz, as applied above, and further in view of Hsu, as applied above.

64. Referring to claim 45, Shiell in view of Dietz has taught a microprocessor as described in claim 31. Furthermore, Shiell in view of Dietz has further taught instruction decode logic (see Fig.1 of Shiell), coupled to said instruction buffer, for determining a length of each of said instructions (See Fig.8 of Hsu and note the actual length ILEN is determined), and prediction check logic, coupled to receive said plurality of indicators from said instruction buffer, wherein if one of said plurality of indicators associated with one of said instructions specifies the microprocessor has speculatively branched to said

one of said branch target addresses of said one of said instructions, said prediction check logic indicates that the microprocessor erroneously branched to said one of said branch target addresses if said length received from said instruction decode logic does not match a speculative length of said one of said instructions provided by said branch target address cache. However, Hsu has taught that if the predicted length results in the system believing that the branch spans multiple cache lines (blocks), then a repeated prediction will be made. See column 13, lines 18-30. Clearly, if the predicted length is wrong then the repeated prediction is erroneous. By caching the length and performing length predictions, the system is able to accommodate branch instructions that cross multiple cache blocks, which is useful in variable length instruction systems, as both Hsu and Shiell are (see the title of Hsu and column 6, lines 32-35 of Shiell). Consequently, it would have been obvious to one of ordinary skill in the art at the time of the invention to modify Shiell to cache the length of a branch instruction so that the system may accommodate branches that span multiple cache lines. However, as explained above, with such a system, erroneous predictions must be detected is the speculative length is incorrect.

Response to Arguments

65. Applicant's arguments filed on June 8, 2004, have been fully considered but they are not persuasive.

66. Applicant argues the novelty/rejection of claim 31 on page 25 of the remarks, in substance that:

> "Applicant respectfully asserts that Dietz does not teach an indicator associated with an instruction in an instruction buffer indicating whether the instruction associated with the

> indicator is a previously executed branch instruction whose target address provided by a branch target address cache has been speculatively branched to by the microprocessor, as recited in amended claim 31. First, Dietz's speculative bit does not indicate whether an instruction associated therewith is a previously executed branch instruction whose target address provided by a branch target address cache has been speculatively branched to by the microprocessor, as recited in amended claim 31. Rather, Dietz's speculative bit indicates whether an instruction, regardless of whether it is a branch or a non-branch instruction, is in a speculative execution path by virtue of a previously predicted taken branch instruction, not that the instruction is a branch instruction that was speculatively predicted taken."
>
> "Furthermore, Dietz teaches that it is instructions in the execution pipeline of his processor (i.e., elements 22, 28, and 30; see col. 3, lines 62-65), which are below the dispatch unit (element 20 of Fig. 1) in the pipeline, that have a speculative bit. This is because until the dispatch unit dispatches the instruction, it cannot be known whether the instruction is in a speculative execution path (see block 66 of Fig. 2). This further implies that instructions in Dietz's instruction queue (element 19 of Fig. 1) cannot have a speculative bit since the instruction queue precedes the instruction dispatch unit, i.e., when the instructions are in the instruction queue it cannot be known yet whether the instructions are in a speculative execution path. Therefore, it would not have been obvious to modify Shiell's instruction buffer to add Dietz's speculative bits thereto since Shiell's instruction buffer (element 60 of Fig. 2, which is included in fetch unit 26 of Fig. 1) also precedes Shiell's instruction scheduler (element 36 of Fig. 1). For each of the reasons stated above, Applicant respectfully asserts that Shiell and Dietz do not obviate claim 31."

67. These arguments are not found persuasive for the following reasons:

a) Regarding the first argument, the examiner asserts that applicant is arguing limitations which are not in the claim. More specifically, applicant is arguing that the indicator bit is associated with previously executed branch instructions (see quoted section above). However, such an association is not established within the claim. Instead, the claim states that indicators are associated with "said instructions" which are instructions that exist within the instruction cache. These instructions do not only have to be previously executed branch instructions.

b) Regarding the second argument, the examiner asserts that it is perfectly acceptable that the speculative bit of Dietz be stored in the instruction buffer of Shiell because when a branch is predicted, the instructions fetched at the predicted target address are speculative instructions. As they are fetched, they would be stored within the instruction buffer and

the speculative bit may be stored with them. In addition, the location of the speculative bit is generally not given patentable weight or would have been an obvious improvement. See In re Japikse 86 USPQ 70 (CCPA 1950).

Conclusion

Any inquiry concerning this communication or earlier communications from the examiner should be directed to David J. Huisman whose telephone number is (703) 305-7811. The examiner can normally be reached on Monday-Friday (8:00-4:30).

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Eddie Chan can be reached on (703) 305-9712. The fax phone number for the organization where this application or proceeding is assigned is 703-872-9306.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

DJH
David J. Huisman
August 25, 2004

EDDIE CHAN
SUPERVISORY PATENT EXAMINER
TECHNOLOGY CENTER 2100